Available online at www.sciencedirect.com
ScienceDirect
Procedia Social and Behavioral Sciences 8(C) (2010) 588–593
Procedia Social and Behavioral Sciences

International Conference on Mathematics Education Research 2010 (ICMER 2010)

A Model of a Mathematics Editor using Intelligent Agent Technology

Teh Noranis Mohd Aris[a,*], Shahrin Azuan Nazeer[b]

[a]*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia*
[b]*Telekom Malaysia Research and Development*

Abstract

To master the mathematics subject, a lot of exercises need to be done. Mathematical problem solving requires writing mathematics equations and symbols to simplify them to get the answer. A step by step guidance is important to make sure that no mistakes occur. This paper presents a study of existing mathematics editor and proposed a web-based model of a mathematics editor using intelligent agent technology based on the Belief, Desires, Intention (BDI) model. The feature to guide the user step by step is incorporated in the proposed model.

***Keywords*:** Mathematics editor; Web-based; Intelligent agent; BDI model

1. Introduction

Doing mathematics exercise requires writing mathematical expressions/symbols and simplifying mathematics equations. Students usually do mathematics exercises using pen and paper manually. Various computerized mathematical editors have been developed to help students to practice mathematics exercises in various domains such as algebra, calculus, number theory, trigonometric functions, geometry and integration. In designing computerized mathematical editors the kind of input to the mathematics editor is an important factor to be considered. Traditionally, the keyboard is used to type mathematical equations. Advancement in pen-based computing provides a natural and better way to input mathematical expressions. Pen-based computing refers to a computer user-interface using a pen (or stylus) and tablet, rather than devices such as a keyboard and a mouse (Wikipedia The Free Encyclopedia, 2010). A study have shown that pen-based computing have important implications for the development of future mathematical interfaces, especially in real-time where the goal is to communicate quickly and effectively (Gozli *et al.*, 2009).

Handwriting recognition is one of the techniques for implementing pen-based computing besides pointing/locator input, direct manipulation and gesture recognition techniques. Many techniques have been proposed for mathematics symbols recognition such as Hidden Markov Model, structural matching and neural networks.

* Corresponding author.
E-mail address: nuranis@fsktm.upm.edu.my

1877-0428 © 2010 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.sbspro.2010.12.081

Intelligent Agent is an autonomous entity which observes and acts upon an environment (agent property) and directs its activity towards achieving goals (rational property). In addition, Intelligent Agents are able to learn or use knowledge to achieve their goals. Example of an Intelligent Agent is a very simple reflex machine such as a thermostat, or poses a very complex behaviour such as a human being, as a community of human beings working together towards achieving a goal. Intelligent Agent is just an autonomous computer program or software that carries out task on behalf of users and has some intelligence characteristics. A simple agent program can be defined mathematically as an agent function which maps every possible percepts sequence to a possible action the agent can perform or to a coefficient, feedback element, function or constant that affects eventual actions (Wikipedia The Free Encyclopedia, 2010):

$$f : P^* \rightarrow A$$

The agent program maps every possible percept to an action.

Due to the features of Intelligent Agents, it is possible to integrate them in mathematics editors. Available mathematics editors are non-interactive systems. For example, user needs to input a mathematics expression and the system will give the answer in a form of a value or graph. There are also systems that provide mathematical simulators to explain a solution to a mathematics problem step by step. However, these systems do not provide a mechanism to check the user's mathematical solution step by step and to correct any mistakes done. Intelligent agent is proposed because it is a continuously running program that can sense a users action and acts upon the specified action.

This research was conducted to achieve the following objectives:

1. To study existing mathematics editors that aids users in problem solving.
2. To propose a new mathematics editor model by integrating Intelligent Agent technique.

This paper consists of four sections, part one is the introduction. Part two compares available mathematics editors based on the architecture and techniques applied. Part three proposed a conceptual model of a mathematics editor using Intelligent Agent technique. Lastly, part four is the conclusion and future works.

2. Related Research

Several mathematics system selected for this research work consists of WebMath (Vuong *et al.*, 2010), Conic Section Simulator (Bacca *et al.*, 2009), OO-CAS (Hussain *et al.*, 2008), Extended OMDOC (Dietrich *et al.*, 2008), MathLang (Kamareddine & Wells, 2008), JaBIT (Oertal, 2008) and ActiveMath (Melis & Siekmann, 2004).

2.1 WebMath

WebMath is a web-based handwritten mathematics system for supporting mathematical problem solving. The system is based on client-server architecture. It consists of four major components: a standard web server, handwritten mathematical expression editor, computation engine, and a web browser with Ajax-based communicator. The handwriting mathematical expression editor adopts a progressive recognition approach for dynamic recognition of handwritten mathematical expression. This approach consists of three processes: progressive expression partitioning, symbol recognition and progressive structural analysis. The symbol recognition applies the conventional elastic matching algorithm and comprises of two processes: sample symbol generation (aims to generate a database of sample symbols to be used for elastic matching purposes) and recognition (symbol are pre-processed and features are extracted). The computation engine supports mathematical functions such as algebraic simplification and factorization, integration and differentiation. The web browser provides a user-friendly interface for accessing the system using advanced Ajax-based communication.

2.2 Conic Section Simulator

The conic section simulator is a web-based educational mathematics simulator to teach the topic of hyperbola, parabola, circumference and ellipse in rectangular coordinates. It was designed using a multi-agent system and semantic web, an ontology in order to increase the application performance and the feedback to students in web learning environments. The pedagogical design was based on the constructivism pedagogical model and the problem based learning. The educational simulator consists of three main components: structural component, pedagogical and didactical component and technological component. The structural component consists of the following systems: registry, didactic, simulation monitoring, theory and help that perform definite and specific functions and all of them are connected. The pedagogical and didactical component is based on the constructivist model and problem-based learning (PBL). The technological component consists of two intelligent agents: simulation and monitor. The monitor agent is to validate and handle student's action, when the student is going to simulate a conic section, the monitor agent get the equation typed by the student and send a message to the simulation agent asking for the conic section to be draw. The simulation agent receives the message and queries the simulator's ontology which is the knowledge base of the agent using the inference engine of JENA semantic web framework. Rules in the inference engine are executed using backward chaining and forward chaining.

2.3 OO-CAS

OO-CAS provides an object oriented design framework that, include a very simple and interactive user Graphical User Interface (GUI) support for a formula editor, making it a self contained system. The formula editor provides a real-time syntax checking for expressions. OO-CAS consists of three classes: AueCASApplication, Parser and Kernel. The AueCASApplication contains the main method which corresponds to the application entry-point. It also contains the main GUI and it is thus interface class which receives the input string entered by the user in the TextBox GUI component. The process of interpretation thus begins in AueCASApplication where the input-string is delivered to the Parser class. Then, a parse-tree is created. The created parse-tree is sent to the Kernel instance where the first pre-order traversal is performed on it.

2.4 Extended OMDOC

Extended OMDOC supports theory repositories with acyclic theory dependencies, axioms, simple definitions and assertions. It is an integration of the proof assistance system ΩMEGA with the standard scientific text editor T_EX_{MACS}. Mathematical document is written entirely inside the text editor in a controlled language with formulas in LAT_EX style. The notation specified in such a document is used for both parsing and rendering formulas in the document. Hybrid parsing technique is proposed which is suitable for real-time application. Incremental method is proposed for verification. Extended OMDOC consists of three main components: mediator, proof assistance system and semantic repository. The mediator is to preserve consistency between the text editor and the proof assistance system by incrementally propagating changes. The proof assistance system is to maintain the formal object in a way that it is verifiable and such that at the same time a human readable presentation can be extracted. The semantic repository is to store and maintain the mathematical knowledge using structural semantic mark-up or scripting languages, including possibilities to search and retrieve knowledge and access control.

2.5 MathLang

MathLang aim to develop an approach for computerizing mathematical text and knowledge which is flexible enough to connect the different approaches to computerization, which allows various degrees of formalization, and which is compatible with different logical frameworks (e.g. set theory, category theory, type theory, etc.) and proof systems. MathLang supports entry of mathematical text editor in an XML format or using T_EX_{MACS} editor. Methods are provided for adding, checking, and displaying various information aspects. One aspect is a kind of weak type system that assigns categories (term, statement, noun (class), adjective (class modifier, etc.) to parts of the text, deals with binding names to meanings, and checks that a kind of grammatical sense is maintained. Another aspect allows weaving together mathematical meaning and visual presentation and can associate natural language text with its mathematical meaning. Another aspect allows identifying chunks of text, marking their roles (theorem, definition, explanation, example, section, etc.), and indicating relationships between the chunks (A uses B, A contradicts B, A follows from B, etc.). Software tool support can use this aspect to check and explain the overall logical structure of a text. Further aspects are being designed to allow adding additional formality to a text such as proof structure and details of how a human readable proof is encoded into a fully formalized version. A number of mathematical texts have been computerized, helping with the development of these aspects, and indicating what additional work is needed for the future.

2.6 ActiveMath

ActiveMath is a web-based intelligent tutoring system for mathematics. ACTIVEMATH that applies Artificial Intelligence (AI) techniques consists of a pedagogical rule base educational content MBase and the student model. ActiveMath has a client-server architecture whose client can be restricted to a browser. On the client side, a browser is sufficient to work with ActiveMath. On the serverside components of ActiveMath have been designed in a modular way. When the user chooses the goal concepts and learning scenario, the session manager sends a request to the course generator. The course generator is responsible for choosing and arranging the content to be learned. The course generator contacts the mathematical knowledge base in order to fetch the identifiers of the mathematical concepts that are required for understanding the goal concepts, queries the student model in order to find out about the user's prior knowledge and preferences, and uses pedagogical rules to select, annotate, and arrange the content – including examples and exercises – in a way that is suitable for the learner. The resulting instructional graph, a list of IDs, is sent to the presentation engine that retrieves the actual mathematical content corresponding to the IDs and that transforms the XML-data to output-pages which are then presented via the user's browser. The course generator and the suggestion mechanism work with the rule based system Jess that evaluates the (pedagogical) rules in order to decide which particular adaptation and content to select and which actions to suggest. Jess uses the Rete algorithm for optimization. External systems such as the computer algebra systems Maple and MuPad and the proof planner Multi are integrated with ActiveMath. They serve as cognitive tools and support the learner in performing complex interactive exercises and they assist in producing feedback by evaluating the learner's input. Also, a diagnosis is passed to the student model in order to update the model.

The discussed mathematical systems are summarized in Table 1.

Table 1. Selected mathematical systems

	Name of Mathematics System	Input Type	Architecture	Technique	Mathematics function support
1	WebMath	Handwritten	Web-based	Progressive expression partitioning, symbol recognition and progressive structural analysis	Algebraic simplification and factorization, integration and differentiation
2	Conic Section Simulator	Keyboard	Web-based	Multi-agent system and semantic web	Teach the topic of hyperbola, parabola, circumference and ellipse in rectangular coordinates
3	OO-CAS	Keyboard	Stand alone	Real-time syntax checking, parsing	Algebra
4	Extended OMDOC	Keyboard	Stand alone	Hybrid parsing, incremental method, semantic mark-up	Theory repositories with acyclic

					theory dependencies, axioms, simple definitions and assertions
5	MathLang	Keyboard	Stand alone	Grammatical checking, visual presentation, associate natural language text association with mathematical meaning	Set theory, category theory, type theory, and proof systems
6	ActiveMath	Keyboard	Web-based	AI techniques, rule base and student model	Algebra

3. Proposed Model

Based on the study that has been done, an innovative web-based mathematical editor that receives handwritten input and combined with intelligent agent technology is suggested. The agent should be able to check each step of the user's solution of a mathematics problem. If any errors exist, the agent should interactively give feedback to the users and guide them by giving suggestions until the correct solution is produced.

The general proposed mathematics editor model is shown in Figure 1. The suggested model adopts the Beliefs, Desires and Intention (BDI) (Wikipedia The Free Encyclopedia, 2010) standard model with modifications in the learning process due to its limitations to learn from past behaviour and adapt to new situations. BDI is a software model developed for programming intelligent agents. Belief represents the informational state of the agent or beliefs about the world. In this case, the belief is the mathematical syntax that needs to be followed when writing mathematical expressions. The mathematical syntax is stored in a belief database. Desires represent the motivational state of the agent, which are the objectives or situations that the agent would like to accomplish or bring about. Here, the desires are the proposed users handwritten which is the input to the editor that needs to be recognized. Intentions represent the deliberative state of the agent or the execution or action of the agent to accomplish. In this situation, the action to be taken by the agent is to guide the user of the system to produce a solution to the mathematics solution.

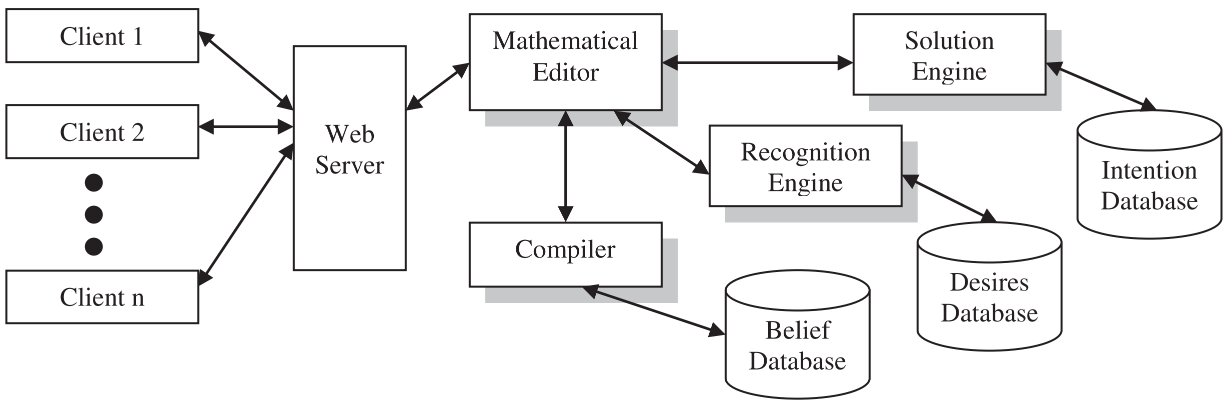

Figure 1: Proposed Mathematics Editor Model

4. Conclusion and Future Works

A study of selected mathematics editors have been done to identify the features of the system in terms of the input type, architecture, techniques and mathematics support function. Based on the study, a proposed mathematics editor model is suggested to integrate intelligent agent technology based on the Beliefs, Desires and Intentions standard model.

For the future works, the proposed model will be implemented using the JADEX intelligent agent framework and Apache Tomcat web server.

References

Bacca, J. L., Molina, J. A., Cavanzo, G. A., & Guevara, J. C. (2009). Educational Simulator of Conic Sections Supported by İntelligent Agents And Semantic Web: An Example in the Development and Design of Educational Simulators for Mathematics. *Frontiers in Education Conference, 2009*, 1-6.

Dietrich, D., Schulz, E., & Wagner, M. (2008). Authoring Verified Documents by İnteractive Proof Construction and Verification in Text-editors. *Lecture Notes in Computer Science*, 398-414.

Gozli, D. G., Pollanen, M., & Reynolds, M. (2009). The Characteristics of Writing Environments for Mathematics: Behavioral Consequences and İmplications for Software Design and Usability. *Lecture Notes in Computer Science*, 310-324.

Hussain, A. D. M., Haq, H. A., Khan, Z. U., & Ahmed, Z. (2008). Simple Object Oriented Designed Computer Algebra System. *Journal of Computational Methods in Sciences and Engineering*, *8*(3), 195-211.

Kamareddine, F., & Wells, J. B. (2008). Computerizing Mathematical Text with MathLang. *Electronic Notes in Theoretical Computer Science*, 25-30.

Melis, E., & Siekmann, J. (2004). ACTIVEMATH: An İntelligent Tutoring System for Mathematics. *Lecture Notes in Artificial Intelligence*, 91-101.

Oertel, M., Himmelspach, J., & Martens, A. (2008). Teaching and Training System Plus Modeling and Simulation- A Plug-İn Based Approach. *Proceedings of UKSim 10th International Conference on Computer Modelling and Simulation*, 475-480.

Vuong, B. –Q, He, Y., & Hui, S. C. (2010). Towards a Web-Based Progressive Handwriting Recognition Environment for Mathematical Problem Solving. *Expert Systems with Applications*, *37*(1), 886-893.

Wikipedia The Free Encyclopedia (2010). *Belief-Desire-Intention software model*. Retrieved August 1, 2010, from http://en.wikipedia.org/wiki/Belief-Desire-Intention_software_model.

Wikipedia The Free Encyclopedia (2010). *Intelligent agent*. Retrieved August 1, 2010, from http://en.wikipedia.org/wiki/Intelligent_agent.

Wikipedia The Free Encyclopedia (2010). *Pen Computing*. Retrieved August 1, 2010, from http://en.wikipedia.org/wiki/Pen_computing.